JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 08, ISSUE 03, 2021

Android Developers Access Installed Apps On The Device Of The Users

Sunny Arora[1], Jatinder Singh Bal[2]
[1,2]Guru Kashi University, Talwandi Sabo

Abstract

Researchers recently discovered a major privacy risk in Android in an extensive report. To get a list of other apps installed on a user device, Android apps utilise Google's IAMs (Installed Application Methods).

This allows for app interoperability, as developers can query for the list of apps that have been installed on a given device using installed application methods (IAMs) on the Android platform. Knowing that IAM data can be used to accurately deduce end-user interests and personality traits raises questions about privacy and data security for everyone. Here, we report on a large-scale empirical investigation into the use of IAMs by Android developers and the prevalence of IAMs in Android apps.

Millions of people have been able to connect to the internet for the first time in the last few years thanks to cheap smart phones. It is possible for Android phone manufacturers to include pre-installed apps (sometimes referred to as "bundled apps" or "bloatware") because the operating system is open-source.

Keywords: Android, Developers, Apps, IAM.

1. Introduction

Installed application methods were originally designed by Google to allow developers to get specific information about other apps installed on a user's device to check for incompatibilities or to improve their own applications by tweaking some features.

The problem, however, is that some Android apps misuse these API calls to gather a list of installed apps and then sell it to advertisers, according to the research. For example, a user's gender, religious beliefs, languages spoken or age group can all be determined by analysing other apps installed on their phone. As a result, Android users face a serious privacy threat.

Researchers from Italy, the Netherlands and Switzerland conducted the study. A large number of popular Android apps and their code were analysed by researchers, and IAM API calls were found. All 14,342 top-category apps and another 7,886 apps whose source codes were publicly available are included in this collection of 14,342 Android applications.

Over 4,214 of the 14,342 apps analysed used IAM calls in their code, according to the analysis. This accounts for more than a third of the most popular apps. Only 2.89 percent of those whose source code has already been made public use these API calls.

What's worse is that, because IAM-based fingerprinting is a "silent method," users cannot even protect themselves from this privacy risk. You don't have to give permission to run codes on your device if an app uses these API calls. Even when developers aren't aware, IAM calls are sometimes made.

In order to create feature-rich apps that take full advantage of the device and platform capabilities, the Android platform provides a wide range of APIs to application developers. App interoperability is made possible in part by APIs that make it possible to retrieve various pieces of data related to the apps that are currently running on the device. These methods are silent from the perspective of the user because no special authorization is needed to use them and no visual indication is provided while they are operating. As a result, most people are unaware that these methods exist. These methods will be referred to as Installed Application Methods in the future (IAMs).

There are currently more than 60 apps installed on the average smartphone user's device based on her interests and personal characteristics (e.g., gender, spoken languages, religious beliefs). Developers may wonder how much information about users can be gleaned from the list of installed applications, which is readily available to them. Many of these characteristics can be inferred with near-optimal accuracy, as discussed in Section 2. As a result, IAMs raise questions about their impact on personal privacy.

IAMs are prevalent in Android apps, but no investigation has been done into their prevalence and how they are used by Android developers. The purpose of our paper is to fill in this knowledge gap by examining how Android developers make use of IAMs. We're here to get a sense of how widely IAMs are being used and to shed some light on why that is.

2. Android (Operating System)

For touchscreen mobile devices such as smartphones and tablets, Android uses a modified version of the Linux kernel and other free and open source software. The Open Handset Alliance, a group of developers, and Google work together to create Android. When it was first announced in November 2007, the HTC Dream was the first commercial Android device.

The majority of Android versions are not open source. The Android Open Source Project (AOSP) provides the foundation for this project, which is primarily licenced under the Apache License for free and open-source software (FOSS). In most cases, when Android is installed on a mobile device, the ability to modify the otherwise FOSS software is restricted, either by not providing the corresponding source code or preventing reinstallation through technical measures, making the installed version proprietary. Google Mobile Services (GMS)[14] includes core apps like Google Chrome, the digital distribution platform Google Play, and the associated Google Play Services development platform, and is pre-installed on the majority of Android devices.

2.1. Development of Android:

The Android Open Source Project (AOSP), an open source initiative led by Google, is responsible for developing Android until the most recent changes and updates are ready to be released. All Android One and Nexus devices have a minimal amount of modifications to the Android operating system's source code.

Original equipment manufacturers (OEMs) customise the source code to run on their hardware. Device drivers, which are often proprietary, are not included in the Android source code. This has resulted in a mixed bag of free and open source and proprietary software on most Android devices, including Google's own.

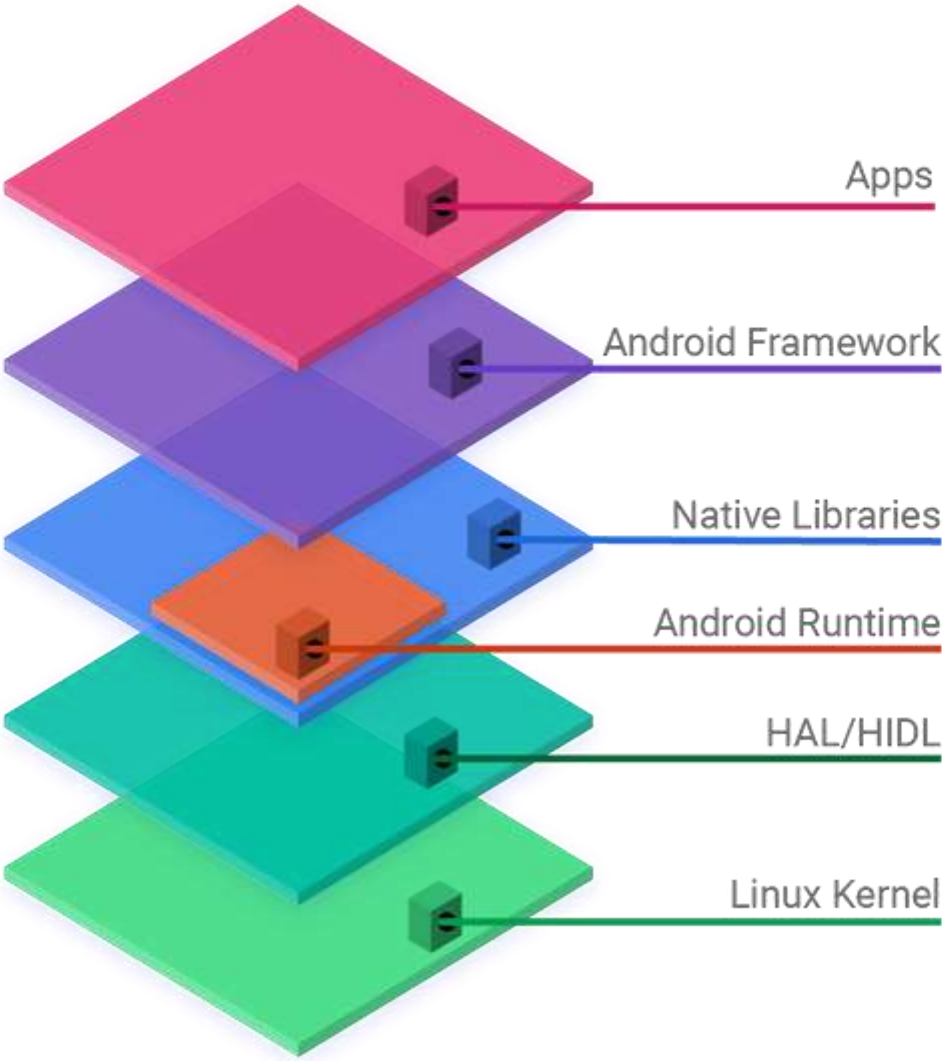

Figure: 1. The Stack of Android Open Source Project.

3. Related Work

An investigation by Seneviratne et al. on user profiling via IAMs was the first. Similarly, Malmi et al. [25] and Frey et al. [16] conducted similar research. According to Demetriou et al., the extent to which IAMs' information can be used by advertising libraries has been examined. Section 2 provides an in-depth examination of these works. Based on their findings, we hope to paint a more complete picture of the spread and actual application of IAMs. Zhou et al. combined data from several seemingly innocent Android APIs, including IAMs, to infer personal information like the gender and religion of users [54].

Using IAMs in the context of mobile health apps presents a significant privacy risk, as their mere presence on the user device can expose particularly sensitive information. Hide MyApp [30] was created by Pham and colleagues to address this problem. It uses user-level virtualization techniques to hide the presence of sensitive apps. HideMyApp, when tested, had a minimal impact on performance and was well-received by end-users.

Somewhat unrelated to our work are investigations into possible methods for the identification of users in a unique way. Historically, companies have used browser fingerprinting to identify unique visitors to their websites. A number of studies have looked at various methods for app fingerprinting, including network traffic patterns, power consumption, memory footprints and UI states.

4. Security and Privacy

Android Partner Vulnerability Initiative was launched by Google in 2020 as a means of enhancing Android's security. Additionally, they formed an Android security team.

4.1. Common security threats

An analysis by security firm Trend Micro shows that premium service abuse, in which infected phones send texts to premium-rate phone numbers without the user's knowledge or consent, is by far the most common type of Android malware. Others display intrusive advertisements or send personal information to third parties that are not authorised by the user. Google engineers have argued that the threat of malware and virus on Android is exaggerated by security companies for commercial gain, and have accused the security industry of playing on fears to sell virus protection software to users. It is claimed by Google that dangerous malware is extremely rare, and a survey by F-Secure found that only 0.5 percent of Android malware reported had come from the Google Play store.

Reporters and researchers discovered in 2021 that a private company had developed and distributed a spyware called Pegasus that could be used to infect both iOS and Android smartphones without any user-interaction or significant clues and then be used to extract data, track user locations, capture film through its camera, and activate the microphone at any time. Pegasus was discovered by journalists and researchers in 2021. An investigation into the data traffic of popular smartphones running Android variants found significant data collection and sharing by default, with no option to opt out. A security patch does not address or cannot address these issues.

4.2. Location-tracking

Wi-Fi access points encountered by Android phone users can be reported to build databases with the physical locations of hundreds of millions of such access points. To run apps such as Foursquare, Google Latitude, and Facebook Places, and to deliver location-based ads, these databases form electronic maps. Third-party monitoring software like TaintDroid, an academic research-funded project, can, in some cases, detect when personal information is being sent from applications to remote servers.

4.3. Technical security features

For pre-installed apps, the user may not be able to explicitly grant access permissions to the rest of the system's resources unless the user explicitly grants access permissions when the application is installed. For example, turning off the pre-installed camera app's microphone access necessitates also turning off the camera. This holds true for Android 7.0 and Android 8.0.

To ensure that Google Play apps are safe, the company has been using its Google Bouncer malware scanner since February 2012. The Android 4.2 "Jelly Bean" operating system version introduced a "Verify Apps" feature in November 2012 to scan all apps, both from Google Play and from third-party sources, for malicious behaviour. This feature has since been removed. After a 2014 update, Verify Apps began scanning apps on a "constant" basis, and the feature was made available to users in the Settings menu in 2017.

5. Conclusion

Analyzing IAM usage in 14,342 free Google Play Store apps and 7,886 opensource Android apps, we conducted a large-scale empirical study to find out how IAMs are actually used. In order to get a better understanding of user privacy concerns, we conducted an investigation into how IAM is used in these apps. In addition, we distributed an online survey to app developers to gauge their level of familiarity with IAMs. Based on our findings, we came up with a list of recommendations for improving the Android platform so that developers and end-users alike are more aware of and in control of IAMs. Among the tasks ahead are examining the ways in which developers are making use of the IAMs fields, perhaps by reaching out to Android developers directly, as well as investigating the rate at which IAMs adoption has increased over time and, finally, developing new solutions that allow for app interoperability while protecting the privacy of end users.

6. References

[1] Kevin Allix, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. 2016. Androzoo: Collecting millions of android apps for the research community. In 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR). IEEE, 468–471.

[2] App Annie. 2017. Spotlight on Consumer App Usage, Part 1.

[3] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. 2014. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. In Acm Sigplan Notices, Vol. 49. ACM, 259–269.

[4] Qi Alfred Chen, Zhiyun Qian, and Z Morley Mao. 2014. Peeking into Your App without Actually Seeing It:{UI} State Inference and Novel Android Attacks. In 23rd {USENIX} Security Symposium ({USENIX} Security 14). 1037–1052.

[5] Yimin Chen, Xiaocong Jin, Jingchao Sun, Rui Zhang, and Yanchao Zhang. 2017. POWERFUL: Mobile app fingerprinting via power analysis. In IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, 1–9.

[6] Michael L Collard, Michael J Decker, and Jonathan I Maletic. 2011. Lightweight transformation and fact extraction with the srcML toolkit. In 2011 IEEE 11th international working conference on source code analysis and manipulation. IEEE, 173–184.

[7] European Commission. 2018. General Data Protection Regulation.

[8] Thomas D Cook, Donald Thomas Campbell, and Arles Day. 1979. Quasiexperimentation: Design & analysis issues for field settings. Vol. 351. Houghton Mifflin Boston.

[9] Shuaifu Dai, Alok Tongaonkar, Xiaoyin Wang, Antonio Nucci, and Dawn Song. 2013. Networkprofiler: Towards automatic fingerprinting of android apps. In 2013 Proceedings IEEE INFOCOM. IEEE, 809–817.

[10] Alexandre Decan, Tom Mens, and Eleni Constantinou. 2018. On the impact of security vulnerabilities in the npm package dependency network. In 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR). IEEE, 181–191.

[11] "32-bits is dead: Here's what it means for Android, Apple, and more". Android Authority. June 12, 2021. Retrieved November 22, 2021.

[12] ^ "android/platform/bionic/". Archived from the original on December 3, 2017.

[13] ^ "android/platform/external/mksh/". Archived from the original on January 21, 2016.

[14] ^ "android/platform/external/toybox/toys/". Archived from the original on March 14, 2016.

[15] ^ "Licenses". Android Source. Archived from the original on December 15, 2016. Retrieved March 11, 2017.